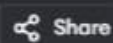




main.c



Run

Output


Clear

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define TABLE_SIZE 10
6
7 typedef struct {
8     char key[20];
9     int value;
10 } HashEntry;
11
12 typedef struct {
13     HashEntry* table[TABLE_SIZE];
14 } HashTable;
15
16 unsigned int hashFunction(char* key) {
17     unsigned int hash = 0;
18     for (int i = 0; i < strlen(key); i++) {
19         hash += key[i];
20     }
21     return hash % TABLE_SIZE;
22 }
23
24 void initHashTable(HashTable* table) {
25     for (int i = 0; i < TABLE_SIZE; i++) {
26         table->table[i] = NULL;
```

```
/tmp/31FF7EzHox.o
Index 0: apple:10
Index 1: mango:50
Index 2: Empty
Index 3: Empty
Index 4: Empty
Index 5: Empty
Index 6: orange:30
Index 7: grape:40
Index 8: Empty
Index 9: banana:20
```

=== Code Execution Successful ===



 Share













Output

Clear

```
Index 0: apple:10
Index 1: Empty
Index 2: Empty
Index 3: Empty
Index 4: Empty
Index 5: Empty
Index 6: orange:30
Index 7: grape:40
Index 8: mango:50
Index 9: banana:20
```

```
=== Code Execution Successful ===
```





main.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define TABLE_SIZE 10
6
7 typedef struct {
8     char key[20];
9     int value;
10 } HashEntry;
11
12 typedef struct {
13     HashEntry* table[TABLE_SIZE];
14 } HashTable;
15
16 unsigned int hashFunction(char* key) {
17     unsigned int hash = 0;
18     for (int i = 0; i < strlen(key); i++) {
19         hash += key[i];
20     }
21     return hash % TABLE_SIZE;
22 }
23
24 void initHashTable(HashTable* table) {
25     for (int i = 0; i < TABLE_SIZE; i++) {
26         table->table[i] = NULL;
```

Output

```
/tmp/nLiYXNg0Pw.o
Index 0: banana:20
Index 1: apple:10
Index 2: Empty
Index 3: Empty
Index 4: mango:50
Index 5: Empty
Index 6: Empty
Index 7: orange:30
Index 8: grape:40
Index 9: Empty

=== Code Execution Successful ===
```

Clear

main.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define TABLE_SIZE 10
6
7 typedef struct Node {
8     char key[20];
9     int value;
10    struct Node* next;
11 } Node;
12
13 typedef struct {
14     Node* table[TABLE_SIZE];
15 } HashTable;
16
17 unsigned int hashFunction(char* key) {
18     unsigned int hash = 0;
19     for (int i = 0; i < strlen(key); i++) {
20         hash += key[i];
21     }
22     return hash % TABLE_SIZE;
23 }
24
25 void initHashTable(HashTable* table) {
26     for (int i = 0; i < TABLE_SIZE; i++) {
```



Share

Run

Output

Clear

```
/tmp/.JvXdt0bAZ.o
Index 0: apple:10 -> mango:50 ->
Index 1:
Index 2:
Index 3:
Index 4:
Index 5:
Index 6: orange:30 ->
Index 7: grape:40 ->
Index 8:
Index 9: banana:20 ->
```

=== Code Execution Successful ===